유전자 알고리즘을 이용한 예측제어

신 승 철, 변 증 남
한국과학기술원 전기및전자공학과
전화: (042) 869-8751; 팩스: (042) 869-8750

Constrained GA-based Predictive Control

Seung C. Shin and Zeungnam Bien
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon, 305-701, KOREA
(E-mail:ssc@ctrsys.kaist.ac.kr)

Abstract

A GA-based optimization technique is adopted in the paper to obtain optimal future control inputs for predictive control systems. For reliable future predictions of a process, we identify the underlying process with an NNARX model structure and investigate to reduce the volume of neural network based on the Lipschitz index and a criterion. Since most industrial processes are subject to their constraints, we deal with the input-output constraints by modifying some genetic operators and/or using a penalty strategy in the GAPC. Some computer simulations are given to show the effectiveness of the GAPC method compared with the adaptive GPC algorithm.

I. Introduction

Usually, the neural networks (NN) used in the methods of predictive control are trained to learn the structure of the underlying process and used to derive a control action [1]-[2]. The nonlinearity of neural networks thus rendered is amenable an optimization technique, which may lead to utilization of nonlinear programming methods. However, the loacl minimum problem of these methods is still outstanding and needs to be resolved by some new methodology such as genetic algorithm (GA).

In the paper, we adopt the GA-based optimization technique to obtain optimal future control inputs, noting that the GA-method is known to have better opportunities for finding an optimal value than a descent-based nonlinear programming method for optimization problems [3].

The GA-based control problem has been formulated in [4], with an assuming that all signals possess unlimited bounds, but such an assuption is unrealistic because in practice all processes are subject to constraints. Therefore, we propose to deal with the input-output constraints of the processes by modifying genetic operators and/or using a penalty strategy. Some simulations for nonlinear processes are given to show the superiority of the GA-based predictive control (GAPC) method to the adaptive GPC algorithm.

II. NN-based System Identification

In this section, we describe the procedure of system identification using a neural network-based autoregressive with extra input (NNARX) model.

To identify the characteristics of a plant, we assume that experimental data describing the underlying system in its entire operating region has been obtained beforehand with a proper choice of sampling frequency. Using the available input-output data pairs, we construct an NNARX model [5]to identify the relationship between input and output of the plant. The regressor vector in the NNARX model can be described as follows:

$$\begin{aligned}\phi(t) &= [y(t-1)\ \cdots\ y(t-n_a)\ u(t-n_k) \\ &\qquad \cdots\ u(t-n_k-n_b+1)]^T\end{aligned} \tag{1}$$

and the predictor can be written as

$$\hat{y}(t|\theta) = \hat{y}(t|t-1,\theta) = g(\phi(t),\theta) \tag{2}$$

where θ is a vector of parameters and g designates the function realized by the neural network. n_a and n_b denote the model order and n_k is the delay time of the system. Fig. 1 illustrates the structure of NNARX model.

The quality of predictions in a model deeply depends on how to select the model order and the delay time in the regressor vector $\phi(t)$, hence it is very important to choose them accurately. To identify the

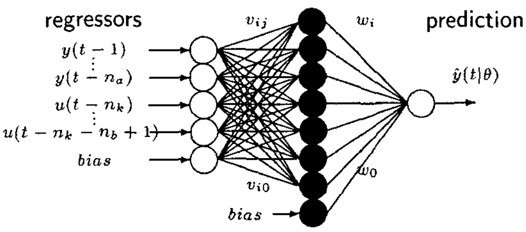

Fig. 1: NNARX model structure

exact model order, n_a and n_b, of a system, we examine the Lipschitz index defined in [6]. In addition, to estimate the exact delay time by using only the data pairs allowed, we consider a simple low order model and evaluate a criterion, such as mean square error (MSE) according to the value of delays.

Using the estimated model order and delay time, we build the regressor vector in (1), and estimate the parameter vector θ in (2) by a prediction error approach minimizing the following MSE criterion:

$$V_N(\theta, Z^N) = \frac{1}{2N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t|\theta)\right)^2. \quad (3)$$

Since the objective function (3) is related to θ nonlinearly, we find the estimate $\hat{\theta}$ by some iterative minimization scheme of the form:

$$\theta_{k+1} = \theta_k + \mu_k d_k \quad (4)$$

where θ_k specifies the parameter vector, d_k is the search direction, and μ_k is the step size at k-th iteration. Because of its rapid convergence properties and robustness, we select the Gauss-Newton based Levenberg-Marquardt method to find $\hat{\theta}$.

After estimating the parameters of the neural network, a validation process should be introduced for the model built. If the constructed NNARX model satisfies given specifications, we then use it as a predictor in the control system. If not, we should follow the feedback paths in the procedure of system identification to try another one.

III. Unconstrained GAPC

Based on the NNARX model describing the given plant precisely, we design a genetic algorithm-based predictive control system in this section.

We assume that the plant can be described by a discrete-time nonlinear dynamical system as follows:

$$\begin{aligned} y(t) &= f\big(y(t-1), \cdots, y(t-n_a), u(t-n_k), \\ &\quad \cdots, u(t-n_k-n_b+1)\big) \end{aligned} \quad (5)$$

where u and y are the input and output of the plant respectively, and $f : R^{n_a+n_b} \to R$ is an unknown smooth function. For the plant in (5), we construct an NNARX model to identify the function f. From the neural network-based identifier, we obtain the following n-step-ahead predictions of the plant:

$$\hat{y}(t+n) = \sum_{i=1}^{h} w_i \sigma\left(\sum_{j=1}^{m} v_{ij} z_j^n + v_{i0}\right) + w_0 \quad (6)$$

where z_j^n is the j-th element of the vector $Z^n = [\hat{y}(t+n-1) \cdots \hat{y}(t+n-n_a)\ u(t+n-n_k) \cdots u(t+n-n_k-n_b+1)]^T$, $m = n_a + n_b$ denotes the number of input nodes except the bias node, $bias = 1$, h is the number of hidden neurons, and σ denotes a hyperbolic activation function.

In the predictive control, the objective is to find the control action, $\underline{u} = [u(t) \cdots u(t+N_u-1)]^T$, which minimizes the following quadratic cost function:

$$J(\underline{u}) = \sum_{n=N_1}^{N_2} e(t+n)^2 + \lambda \sum_{n=0}^{N_u-1} \Delta u(t+n)^2 \quad (7)$$

where $e(t+n) = r(t+n) - \hat{y}(t+n)$, N_1 and N_2 are the minimum and maximum prediction horizon, respectively, N_u is the control horizon, λ is the control weighting factor, $\Delta u(t) = u(t) - u(t-1)$, and $r(t)$ is the reference signal.

We implement the genetic algorithm using real-valued string encoding because it takes less search time than the binary string encodig. The population consists of a μ-tuple of candidate solutions, called chromosomes or individuals, $s_i \in \Omega$, $i = 1, 2, \cdots, \mu$. Note that an individual solution s_i represents an arbitrary value of $\underline{u}$, say $\underline{u}^i$, to be optimized in the GA-based predictive control scheme. The quality of an individual is measured by its fitness function $F : \Omega \to R$ defined on the basis of the objective function. We adopt typical crossover and mutation operators to recombine the individuals in the population according to the probability of crossover(p_c) and muation(p_m), respectively.

To show the control performance of the GA-based predictive control, we perform computer simulations compared with adaptive GPC for a plant. In the paper, we set N_1=1, N_2=3, N_u=1, and λ=1.0.

■ EXAMPLE 1: Consider the following nonlinear discrete-time process described by

$$y(t) = \frac{x_1 x_2 x_3 x_5 (x_2 - 1) + x_4}{1 + x_2^2 + x_3^2}$$

where $x_1 = y(t-1), x_2 = y(t-2), x_3 = y(t-3), x_4 = u(t-1)$, and $x_5 = u(t-2)$.

To realize the adaptive GPC algorithm, we construct a linear ARX model by assigning the exact system order to the model order and estimate the parameters of the model using the recursive least square

(RLS) method with forgetting factor ($\gamma = 0.90$) online. In the GAPC, we set μ=50, p_m=0.25, p_c=0.12, and the generation size ρ=20. In addition, we set the input space Ω in the genetic algorithm to [-2.0, 4.0]. By comparing Fig. 2 with 3 , we find that the GA-based control system yields prominent tracking performance, whereas the adaptive GPC makes undesirable large fluctuations near $r(t) = \pm 1$.

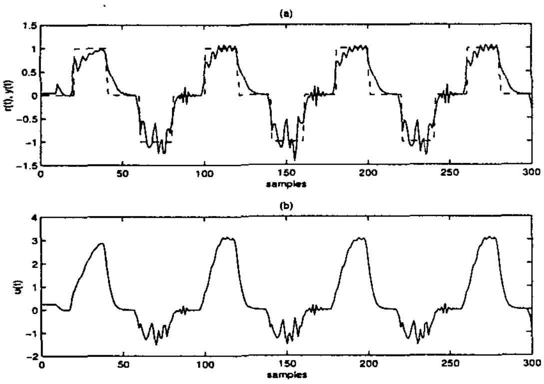

Fig. 2: adaptive GPC results for the plant in EXAMPLE 1

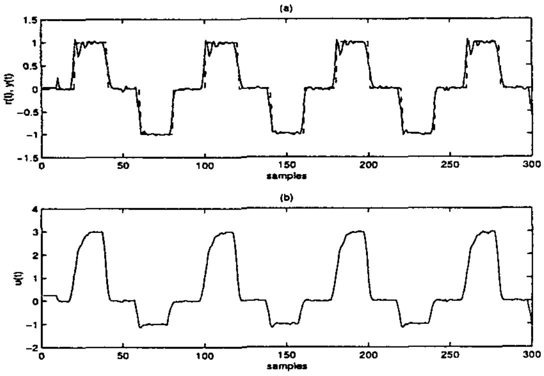

Fig. 3: GAPC results for the plant in EXAMPLE 1

Although the GA-based predictive control method gives good tracking performance, the process time should be taken into account in the real applications. Fig. 4 illustrate some control performances of the GAPC for the plant in EXAMPLE 1.

IV. Constrained GAPC

We have formulated the control problem assuming that all the signals possess unbounded ranges, however, it is not realistic because most practical processes are subject to constraints. Including the amplitude limits of input/output signal, slew rate limits of an actuator, and overshoot constraints of

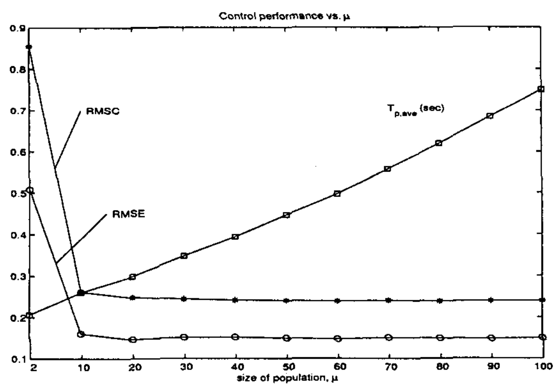

Fig. 4: Control performances of GAPC according to μ for the plant in EXAMPLE 1 ($\rho = 20$)

the output, we find a control vector $\underline{u}$ minimizing the objective function (7) subject to

$$\begin{array}{rcll} u_L \leq & u(t) & \leq u_H & \forall\, t, \\ \Delta u_L \leq & \Delta u(t) & \leq \Delta u_H & \forall\, t, \\ y_L \leq & y(t) & \leq y_H & \forall\, t, \\ & y(t) & \leq r(t) & \forall\, t \end{array} \qquad (8)$$

where the subscripts, L and H, mean the low and high limit values for each variable, respectively. To deal with the constraints (8) in the GAPC, we modify some genetic operators and use a penalty strategy [3] in the following.

When all constraints are linear in the optimization problem, we can simply treat the constraints by using some specified genetic operators which guarantee that all parents and offsprings lie in a feasible solution space $\Omega \subset R^{N_u}$.

To satisfy the feasible condition of a population at an initial step, we select a subset of potential solutions from the space of whole feasible region randomly, and then fill the remaining subset of potential solutions with boundaries of the solution space. And then, we recombine chromosomes to generate feasible offsprings by using some modified genetic operators such as uniform mutation, boundary mutation, non-uniform mutation, simple crossover, and arithmetical crossover according to the probabilities, p_{um}, p_{bm}, p_{nm}, p_{sc}, and p_{ac}, respectively.

In genetic algorithms, the penalty technique is used to keep a certain amount of infeasible solutions in each generation so as to enforce genetic search towards an optimal solution from both sides of feasible and infeasible regions.

Considering the future predictions of a plant, we design a penalty function $P(\underline{u})$ as follows:

$$P(\underline{u}) = \sum_{i=1}^{N} \alpha^i(k) \sum_{j=1}^{M} d_{ij}^{\beta}(\underline{u}). \qquad (9)$$

where $\alpha^i(k) = \alpha^i \times k$, k denotes the generation step, α and β are the parameters used to adjust the scale of penalty value,

$$d_{ij}(\underline{u}) = \begin{cases} 0 & \text{if } \underline{u} \text{ is feasible,} \\ |g_{ij}(\underline{u})| & \text{otherwise} \end{cases} \tag{10}$$

and g_{ij} denotes the j-th constraint of the i-step-ahead prediction, and M represents the number of constraints in one prediction output. To show the validity of the constrained GAPC, we give some simulation results in the following.

■ Example 2: Consider a simple linear plant described by

$$G(s) = \frac{1}{1 + 10s}$$

where the sampling time is set to 0.6 sec.

To begin with, we handle the amplitude constraints of input signal by using the modified genetic operators where we set μ=50, ρ=20, p_{um}=0.12, p_{bm}=0.15, p_{nm}=0.10, p_{sc}=0.25, and p_{ac}=0.25. We can observe how the introduction of the constraints on the manipulated variables has produced a slower closed loop response as was to be expected in Fig. 5 .

In the next place, we apply a penalty technique to the plant to restrict overshoots of the process output. For the penalty function in (9), we let $\alpha^i(k) = \eta \cdot \alpha^i$ where η=100 and α=0.9. By setting $\beta = 1$, we can write $d_{ij}^{\beta}(\underline{u}) = d_i(\underline{u})$ since $M = 1$, and thus, if $\hat{y}(t+i) < r(t+i)$, $d_i(\underline{u})$=0; otherwise, $d_i(\underline{u}) = |\hat{y}(t+i) - r(t+i)|$.

By applying an unconstrained GAPC to the plant, we obtain the control results as shown in Fig. 6 . As can be seen, the output shows a noticeable overshoot (solid line). However, the overshoot has been eliminated when the overshoot constraints are taken into account as shown in the same figure (dashed line).

V. Concluding Remarks

In the paper, we have investigated the GA-based predictive control scheme when the input and output of the system have been constrained. We have dealt with the constraints by modifying some genetic operations for linear constraints and using a penalty strategy for nonlinear constraints. From our extensive simulation studies, it is found that the tracking error and process time should be carefully chosen to satisfy a certain specification in GA-based predictive control systems.

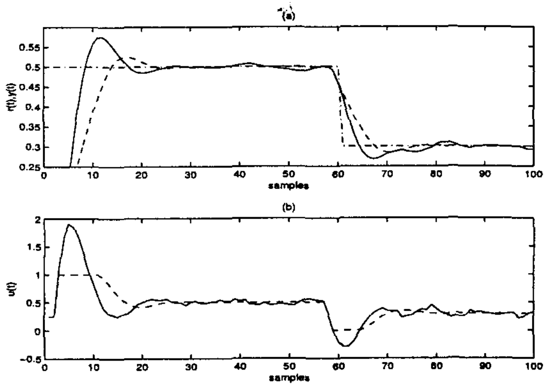

Fig. 5: GAPC with input amplitude constraints

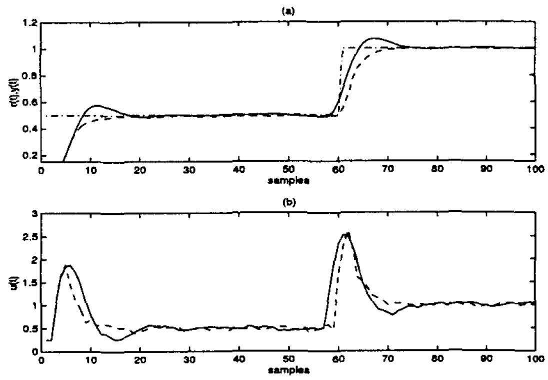

Fig. 6: GAPC with overshoot constraints

References

[1] A. Draeger, S. Engell, and H. Ranke, Model Predictive Control Using Neural Networks, *IEEE Control Systems* **15** (1995) 61-66.

[2] M. Norgaard, PH. Sorensen, N.K. Poulsen, O. Rarn, and L.K. Hansen, Intelligent Predictive Control of Nonlinear Processes Using Neural Networks, *Proc. IEEE Symp. Intelligent Cont.* (1996) 301-306.

[3] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, (John Wiely & Sones, 1997).

[4] S. C. Shin and S. B. Park, GA-based Predictive Control for Nonlinear Processes, *Electronics Letters* **34** (1998) 1980-1981.

[5] J. Sjoberg, H. Hjalmerson, and L. Ljung, Neural networks in system identification, *Proc. IFAC Symp. on SYSID* **2** (1994) 49-71.

[6] X. He and H. Asada, A new method for identifying orders of i-o models for nonlinear dynamical systems, *Proc. American Control Conference* (1993) 2520-2523.